



ATID Application Development Framework Reference Manual –Smart Card

Revision: Ver. 0.1

Date: January, 2012

ATID Co., Ltd.

Table of Contents

Table of Contents	2
Acronym	4
Revision History	5
1 .NET API Reference	6
1.1 Enumerations	6
1.1.1 SCARD_RESULT	6
1.1.2 BAUDRATE	6
1.1.3 OP_MODE	7
1.1.4 VOLTAGE	7
1.1.5 TPROTOCOL	7
1.1.6 STATUS_CODE	7
1.1.7 ERROR_CODE	7
1.2 Methods	9
1.2.1 ProtocolInit	9
1.2.2 ProtocolDeinit	9
1.2.3 PowerOn	9
1.2.4 PowerOff	10
1.2.5 SetParameters	11
1.2.6 GetParameters	12
1.2.7 SetDefaultParameter	12
1.2.8 GetSlotStatus	13
1.2.9 TransferBlock	13
1.2.10 GetFWVersion	14
2 C/C++ API Reference	16
2.1 Enumerations	16
2.1.1 SCARD_RESULT	16
2.2 Constants	17
2.2.1 RATE_115200	17
2.2.2 RATE_1200	17
2.2.3 RATE_14400	17
2.2.4 RATE_19200	17
2.2.5 RATE_2400	17
2.2.6 RATE_28800	17
2.2.7 RATE_38400	17
2.2.8 RATE_4800	17
2.2.9 RATE_57600	17
2.2.10 RATE_600	17

2.2.11	EMV_MODE.....	17
2.2.12	ISO_MODE.....	17
2.2.13	AUTO_VOLTAGE_SELECTION	17
2.2.14	VOLTS_5_0	17
2.2.15	VOLTS_3_0	18
2.2.16	VOLTS_1_8	18
2.2.17	PROTOCOL_T0	18
2.2.18	PROTOCOL_T1	18
2.2.19	BSTATUS_ERROR_MASK.....	18
2.2.20	BSTATUS_ICC_STATUS_MASK	18
2.2.21	BSTATUS_FAILED	18
2.2.22	BSTATUS_TIME_EXT	18
2.2.23	ICC_PRESENT_AND_ACTIVE	18
2.2.24	ICC_PRESENT_AND_INACTIVE	18
2.2.25	ICC_NOT_PRESENT.....	18
2.2.26	ICC_FAILED	18
2.2.27	CMD_ABORTED	18
2.2.28	ICC_MUTE	18
2.2.29	HW_ERROR.....	18
2.2.30	BAD_ATR_TS.....	19
2.2.31	BAD_ATR_TCK.....	19
2.2.32	COMM_TIMEOUT	19
2.2.33	RECEIVED_NAK.....	19
2.2.34	ERROR_NONE	19
2.2.35	CMD_FAILED	19
2.3	Methods.....	20
2.3.1	SCardProtocolInit	20
2.3.2	SCardProtocolDeinit.....	20
2.3.3	SCardPowerOn	20
2.3.4	SCardPowerOff	21
2.3.5	SCardSetParameters.....	22
2.3.6	SCardGetParameters.....	23
2.3.7	SCardSetDefaultParameter.....	23
2.3.8	SCardGetSlotStatus	24
2.3.9	SCardTransferBlock	24
2.3.10	SCardGetFWVersion	25

Acronym

modules	descriptions
AADF	ATID Application Development Framework
ATR	Answer To Reset
APDU	Application Protocol Data Unit
CLA	Class byte
INS	Instruction byte

Revision History

Version	Date	Reason	Description	Author
0.1	2012/01/17	Draft		Y. J. CHO

1 .NET API Reference

1.1 Enumerations

1.1.1 SCARD_RESULT

The result of a call to functions.

- **SCARD_RESULT_SUCCESS**
Function executed successfully.
- **SCARD_RESULT_OUTOFMEMORY**
Failed to assign memory
- **SCARD_RESULT_INVALID_ARGS**
Invalid parameter
- **SCARD_RESULT_UNSUPPORTED**
Not support command currently.
- **SCARD_RESULT_FAILURE**
Command execution failed
- **SCARD_RESULT_ALREADY_OPENED**
SmartCard device already opened
- **SCARD_RESULT_NOT_OPENED**
Call the function without Open
- **SCARD_RESULT_INVALID_DEVICE**
Invalid SmartCard device.

1.1.2 BAUDRATE

Communication speed between PDA and Smart Card

- **RATE_115200**
- **RATE_1200**
- **RATE_14400**
- **RATE_19200**
- **RATE_2400**
- **RATE_28800**
- **RATE_38400**
- **RATE_4800**
- **RATE_57600**

- **RATE_600**
- **RATE_9600**

1.1.3 OP_MODE

Operating mode of Smart Card.

- **EMV_MODE**
- **ISO_MODE**

1.1.4 VOLTAGE

voltage

- **AUTO_VOLTAGE_SELECTION**
- **VOLTS_1_8**
- **VOLTS_3_0**
- **VOLTS_5_0**

1.1.5 TPROTOCOL

Protocol data structure type

- **PROTOCOL_T0**
- **PROTOCOL_T1**

1.1.6 STATUS_CODE

Status of Card slot.

- **BSTATUS_ERROR_MASK**
- **BSTATUS_FAILED**
- **BSTATUS_ICC_STATUS_MASK**
- **BSTATUS_TIME_EXT**
- **ICC_FAILED**
- **ICC_NOT_PRESENT**
- **ICC_PRESENT_AND_ACTIVE**
- **ICC_PRESENT_AND_INACTIVE**

1.1.7 ERROR_CODE

The code result of command execution.

- **BAD_ATR_TS,**
- **BAD_ATR_TCK**
- **CMD_ABORTED,**
- **CMD_FAILED,**
- **COMM_TIMEOUT,**
- **ERROR_NONE,**
- **HW_ERROR,**
- **ICC_MUTE,**
- **RECEIVED_NAK**

1.2 Methods

1.2.1 ProtocolInit

Smart Card devices are required to allocate resources.

```
SCARD_RESULT ProtocolInit();
```

Parameters

None

Return Values

SCARD_RESULT_SUCCESS will be returned if successfully allocate resources.

Notes

It must firstly to be run in order to use Smart Card.

1.2.2 ProtocolDeinit

Deallocate the Smart Card device resources.

```
SCARD_RESULT ProtocolDeinit();
```

Parameters

None

Return Values

SCARD_RESULT_SUCCESS will be returned if successfully deallocate resources.

Notes

When finishing using Smart Card, deallocate resources by all means calling this function.

1.2.3 PowerOn

Apply power to the Smart Card device. after the initial reset is completed, ATR string will be responded. The string consisting of initial letters and 32 additional characters shows the operating properties of the card.

```
SCARD_RESULT PowerOn(  
    byte Slot,  
    OP_MODE Mode,  
    VOLTAGE Powersel,
```

```

    refSTATUS_CODE Status,
    refERROR_CODE Error,
    byte[] pAtr,
    refuint LengthOfAtr
);

```

Parameters

Slot

the slot number which Smart Card is inserted in.

Mode

Smart Card operating mode.

Powersel

Voltage.

Status

Variable that will store status of Smart Card about command Execution.

Error

Variable that will store the execution result about command execution.

pAtr

Byte array which ATR data will be stored in.

LengthOfAtr

The length of ATR data.

Return Values

If performed successfully, SCARD_RESULT_SUCCESS will be returned.

Notes

Refer to ISO-7816 document for more information.

1.2.4 PowerOff

Remove the Smart Card power.

```

SCARD_RESULT PowerOff(
    byte Slot,
    refSTATUS_CODE Status,
    refERROR_CODE Error
);

```

Parameters

Slot

the slot number which Smart Card is inserted in.

Status

Variable that will store status of Smart Card about command Execution.

Error

Variable that will store the execution result about command execution.

Return Values

If performed successfully, SCARD_RESULT_SUCCESS will be returned.

1.2.5 SetParameters

set protocol type, clock rate, bit rate of Smart Card device.

```
SCARD_RESULT SetParameters(  
    byteSlot,  
    T_PROTOCOL T_Protocol,  
    byteFiDi,  
    ref STATUS_CODE Status,  
    ref ERROR_CODE Error  
);
```

Parameters

Slot

the slot number which Smart Card is inserted in.

T_Protocol

Protocol number.

FiDi

Modified value of Clock rate conversion and Bit rate (1byte) (Fi 4bit)XXXX, (Di 4bit)xxxx

Clock rate conversion factor F (XXXX)

FI	0000	0001	0010	0011	0100	0101	0110	0111
FI	Internal clk	372	558	774	1116	1488	1860	RFU
fs (max) MHz	-	5	6	8	12	16	20	-

Bit rate adjustment factor D (xxxx)

DI	0000	0001	0010	0011	0100	0101	0110	0111
D	RFU	1	2	4	8	16	RFU	RFU

DI	1000	1001	1010	1011	1100	1101	1110	1111
D	RFU	RFU	1/2	1/4	1/8	1/16	1/32	RFU

Status

Variable that will store status of Smart Card about command Execution.

Error

Variable that will store the execution result about command execution.

Return Values

If performed successfully, SCARD_RESULT_SUCCESS will be returned.

Notes

Refer to ISO-7816 document for more information.

1.2.6 GetParameters

Get the setting value of protocol type, clock rate, bit rate in Smart Card.

```
SCARD_RESULT GetParameters(
    byte Slot,
    refTPROTOCOLT_Protocol,
    refbyte FiDi,
    refSTATUS_CODE Status,
    refERROR_CODE Error
);
```

Parameters

Slot

the slot number which Smart Card is inserted in.

T_Protocol

protocol number read from Smart card.

FiDi

Setting value of clock rate conversion and bit rate read from Smart Card.

Status

Variable that will store status of Smart Card about command Execution.

Error

Variable that will store the execution result about command execution.

Return Values

If performed successfully, SCARD_RESULT_SUCCESS will be returned.

1.2.7 SetDefaultParameter

Initialize the value of protocol type, clock rate, bit rate in Smart Card device.

```
SCARD_RESULT SetDefaultParameter(
```

```
byte Slot,  
refSTATUS_CODE Status,  
refERROR_CODE Error  
);
```

Parameters

Slot

the slot number which Smart Card is inserted in

Status

Variable that will store status of Smart Card about command Execution.

Error

Variable that will store the execution result about command execution.

Return Values

If performed successfully, SCARD_RESULT_SUCCESS will be returned.

1.2.8 GetSlotStatus

Get the status value of Smart Card slot.

```
SCARD_RESULT GetSlotStatus(  
byte Slot,  
refSTATUS_CODE Status,  
refERROR_CODE Error  
);
```

Parameters

Slot

the slot number which Smart Card is inserted in

Status

Variable that will store status of Smart Card about command Execution.

Error

Variable that will store the execution result about command execution.

Return Values

If performed successfully, SCARD_RESULT_SUCCESS will be returned.

1.2.9 TransferBlock

Transfer APDU Command to Smartd Card

```
SCARD_RESULT TransferBlock(  
byte Slot,
```

```

uintLengthOfTxData,
byte[] TxData,
refuintLengthOfRxData,
byte[] RxData,
refSTATUS_CODE Status,
refERROR_CODE Error
)

```

Parameters

Slot

the slot number which Smart Card is inserted in

LengthOfTxData

Length of data to be transferred.

TxData

byte array which APDU Command is stored in.

LengthOfRxData

The length of data responded from Smart Card, if command execution succeed.

RxData

Response from Smart Card.

Status

Variable that will store status of Smart Card about command Execution.

Error

Variable that will store the execution result about command execution.

Return Values

If performed successfully, SCARD_RESULT_SUCCESS will be returned.

Notes

Format of APDU Command is as follows.

1. CLA – INS – P1 – P2
2. CLA – INS – P1 – P2 – Le
3. CLA – INS – P1 – P2 –Lc– Data
4. CLA – INS – P1 – P2 –Lc– Data – Le

Le field the maximum expected number of bytes of the message and Lc field indicates the length of the Data filed. For more information, see the ISO-7816 document.

1.2.10 GetFWVersion

Get the firmware version from Smart Card device.

```

BOOLGetFWVersion(

```

```
byte Slot,  
byte[] Version,  
refSTATUS_CODE Status,  
refERROR_CODE Error  
)
```

Parameters

Slot

the slot number which Smart Card is inserted in.

Version

Byte array where firmware version of Smart Card device will be stored in.

Status

Variable that will store status of Smart Card about command Execution.

Error

Variable that will store the execution result about command execution.

Return Values

If performed successfully, SCARD_RESULT_SUCCESS will be returned.

2 C/C++ API Reference

2.1 Enumerations

2.1.1 SCARD_RESULT

The result of a call to functions.

- **SCARD_RESULT_SUCCESS**
Function executed successfully.
- **SCARD_RESULT_OUTOFMEMORY**
Failed to assign memory.
- **SCARD_RESULT_INVALID_ARGS**
Invalid parameter.
- **SCARD_RESULT_UNSUPPORTED**
Not support command currently.
- **SCARD_RESULT_FAILURE**
Command execution failed
- **SCARD_RESULT_ALREADY_OPENED**
SmartCard device already opened.
- **SCARD_RESULT_NOT_OPENED**
Call the function without Open.
- **SCARD_RESULT_INVALID_DEVICE**
SmartCard device is not equipped.

2.2 Constants

2.2.1 RATE_115200

Data rate 115200bps

2.2.2 RATE_1200

Data rate 1200bps

2.2.3 RATE_14400

Data rate 14400bps

2.2.4 RATE_19200

Data rate 19200bps

2.2.5 RATE_2400

Data rate 2400bps

2.2.6 RATE_28800

Data rate 28800bps

2.2.7 RATE_38400

Data rate 38400bps

2.2.8 RATE_4800

Data rate 4800bps

2.2.9 RATE_57600

Data rate 57600bps

2.2.10 RATE_600

Data rate 600bps

2.2.11 EMV_MODE

Operating mode of Smartd Card device.

2.2.12 ISO_MODE

Operating mode of Smartd Card

2.2.13 AUTO_VOLTAGE_SELECTION

Voltage to be applied to Smartd Card.

2.2.14 VOLTS_5_0

Voltage to be applied to Smartd Card

2.2.15 VOLTS_3_0

Voltage to be applied to Smartd Card.

2.2.16 VOLTS_1_8

Voltage to be applied to Smartd Card.

2.2.17 PROTOCOL_T0

Protocol data structure type

2.2.18 PROTOCOL_T1

Protocol data structure type

2.2.19 BSTATUS_ERROR_MASK

Mask that detects the value of returned Error.

2.2.20 BSTATUS_ICC_STATUS_MASK

Mask that detects the value of returned status.

2.2.21 BSTATUS_FAILED

Command execution failed.

2.2.22 BSTATUS_TIME_EXT

Command execution failed due to Time Out.

2.2.23 ICC_PRESENT_AND_ACTIVE

Slot Smart Card is inserted, and in an enabled state.

2.2.24 ICC_PRESENT_AND_INACTIVE

Smart Card is inserted, but in an unusable state.

2.2.25 ICC_NOT_PRESENT

No Smart Card in slot.

2.2.26 ICC_FAILED

Checking slow status failed.

2.2.27 CMD_ABORTED

Command execution canceled.

2.2.28 ICC_MUTE

No response from Smart Card.

2.2.29 HW_ERROR

Hardware error.

2.2.30 BAD_ATR_TS

Bad ATR Ts

2.2.31 BAD_ATR_TCK

Bad ATR tck

2.2.32 COMM_TIMEOUT

Transferring command failed, due to Time Out.

2.2.33 RECEIVED_NAK

Receiving NAK from Smart Card.

2.2.34 ERROR_NONE

Transferring command succeed.

2.2.35 CMD_FAILED

Executing command failed.

2.3 Methods

2.3.1 SCardProtocolInit

Smart Card device are required to allocate resources

```
SCARD_RESULT SCardProtocolInit();
```

Parameters

None

Return Values

SCARD_RESULT_SUCCESS will be returned if successfully allocate resources.

Notes

It must firstly to be run in order to use Smart Card device.

2.3.2 SCardProtocolDeinit

Deallocate the Smart Card device resources.

```
SCARD_RESULT SCardProtocolDeinit();
```

Parameters

None

Return Values

SCARD_RESULT_SUCCESS will be returned if successfully deallocate resources.

Notes

When finishing using Smart Card, deallocate resources by all means calling this function.

2.3.3 SCardPowerOn

Apply power to the Smart Card device. after the initial reset is completed, ATR string will be responded. The string consisting of initial letters and 32 additional characters shows the operating properties of the card.

```
SCARD_RESULT SCardPowerOn(  
    UCHAR Slot,  
    UCHAR Mode,  
    UCHAR Powersel,  
    UCHAR* pStatus,  
    UCHAR* pError,
```

```

    UCHAR*pAtr,
    UCHAR*pLengthOfAtr
);

```

Parameters

Slot

the slot number which Smart Card is inserted in.

Mode

Smart Card operating mode.

Powersel

Voltage to be applied to Smart Card.

pStatus

Variable address that will store status of Smart Card about command Execution.

pError

Variable address that will store the execution result about command execution.

pAtr

UCHAR address where ATR data will be stored.

pLengthOfAtr

The length of ATR data.

Return Values

If performed successfully, SCARD_RESULT_SUCCESS will be returned.

Notes

Refer to ISO-7816 document for more information

2.3.4 SCardPowerOff

Remove the Smart Card power.

```

SCARD_RESULTSCardPowerOff(
    UCHAR Slot,
    UCHAR*pStatus,
    UCHAR*pError
);

```

Parameters

Slot

the slot number which Smart Card is inserted in.

pStatus

Variable address that will store status of Smart Card about command Execution.

pError

Variable address that will store the execution result about command execution.

Return Values

If performed successfully, SCARD_RESULT_SUCCESS will be returned.

2.3.5 SCardSetParameters

set protocol type, clock rate, bit rate of Smart Card device

SCARD_RESULT SCardSetParameters (

UCHAR Slot,

UCHAR T_Protocol,

UCHAR FiDi,

UCHAR* pStatus,

UCHAR* pError

);

Parameters

Slot

The number of Smart Card inserted.

T_Protocol

Protocol number.

FiDi

Modified value of Clock rate conversion and Bit rate (1byte) (Fi 4bit)XXXX, (Di 4bit)xxxx

Clock rate conversion factor F (XXXX)

FI	0000	0001	0010	0011	0100	0101	0110	0111
FI	Internal clk	372	558	774	1116	1488	1860	RFU
fs (max) MHz	-	5	6	8	12	16	20	-

Bit rate adjustment factor D (xxxx)

DI	0000	0001	0010	0011	0100	0101	0110	0111
D	RFU	1	2	4	8	16	RFU	RFU

DI	1000	1001	1010	1011	1100	1101	1110	1111
D	RFU	RFU	1/2	1/4	1/8	1/16	1/32	RFU

Status

Variable that will store status of Smart Card about command Execution.

Error

Variable that will store the execution result about command execution.

Return Values

If performed successfully, SCARD_RESULT_SUCCESS will be returned.

Notes

Refer to ISO-7816 document for more information.

2.3.6 SCardGetParameters

Get the setting value of protocol type, clock rate, bit rate in Smart Card.

```
SCARD_RESULT SCardGetParameters (
    UCHAR Slot,
    UCHAR* pT_Protocol,
    UCHAR* pFiDi,
    UCHAR* pStatus,
    UCHAR* pError
);
```

Parameters

Slot

the slot number which Smart Card is inserted in.

pT_Protocol

protocol number read from Smart card.

pFiDi

Setting value of clock rate conversion and bit rate read from Smart Card.

pStatus

Variable address that will store status of Smart Card about command Execution.

pError

Variable address that will store the execution result about command execution.

Return Values

If performed successfully, SCARD_RESULT_SUCCESS will be returned..

2.3.7 SCardSetDefaultParameter

Initialize the value of protocol type, clock rate, bit rate in Smart Card device.

```
SCARD_RESULT SCardSetDefaultParameter (
    UCHAR Slot,
```

```

    UCHAR*pStatus,
    UCHAR*pError
);

```

Parameters

Slot

the slot number which Smart Card is inserted in

pStatus

Variable that will store status of Smart Card about command Execution.

pError

Variable that will store the execution result about command execution.

Return Values

If performed successfully, SCARD_RESULT_SUCCESS will be returned..

2.3.8 SCardGetSlotStatus

Get the status value of Smart Card slot.

```

SCARD_RESULT SCardGetSlotStatus (
    UCHAR Slot,
    UCHAR*pStatus,
    UCHAR*pError
);

```

Parameters

Slot

the slot number which Smart Card is inserted in

pStatus

Variable address that will store status of Smart Card about command Execution.

pError

Variable address that will store the execution result about command execution.

Return Values

If performed successfully, SCARD_RESULT_SUCCESS will be returned.

2.3.9 SCardTransferBlock

Transfer APDU Command to Smartd Card

```

SCARD_RESULT SCardTransferBlock (
    UCHAR Slot,
    UINT LengthOfTxData,

```



```

    UCHAR*pTxData,
    UINT*plengthOfRxData,
    UCHAR*pRxData,
    UCHAR*pStatus,
    UCHAR*pError
)

```

Parameters

Slot

the slot number which Smart Card is inserted in

LengthOfTxData

Length of data to be transferred.

pTxData

UCHAR array which APDU Command is stored in.

plengthOfRxData

The length of data responded from Smart Card, if command execution succeed.

pRxData

Response from Smart Card.

pStatus

Variable address that will store status of Smart Card about command Execution.

pError

Variable address that will store the execution result about command execution.

Return Values

If performed successfully, SCARD_RESULT_SUCCESS will be returned.

Notes

Format of APDU Command is as follows.

5. CLA – INS – P1 – P2
6. CLA – INS – P1 – P2 – Le
7. CLA – INS – P1 – P2 –Lc– Data
8. CLA – INS – P1 – P2 –Lc– Data – Le

Le field the maximum expected number of bytes of the message and Lc field indicates the length of the Data filed. For more information, see the ISO-7816 document.

2.3.10 SCardGetFWVersion

Get the firmware version from Smart Card device.

```

BOOLSCardGetFWVersion(
    UCHAR Slot,

```

```
    UCHAR*pVersion,  
    UCHAR*pStatus,  
    UCHAR*pError  
)
```

Parameters

Slot

the slot number which Smart Card is inserted in.

pVersion

UCHAR array where firmware version of Smart Card device will be stored.

pStatus

Variable address that will store status of Smart Card about command Execution.

pError

Variable address that will store the execution result about command execution.

Return Values

If performed successfully, SCARD_RESULT_SUCCESS will be returned.